Europäisches
Patentamt

Eur pean
Patent Office

Office européen
des brevets

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. **Patent application No.** **Demande de brevet n°**

03100339.5

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

DEN HAAG, DEN
THE HAGUE, 17/02/03
LA HAYE, LE

Eur päisches
Patentamt

Eur pean
Patent Office

Office européen
des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:
Demande n°:
03100339.5

Anmeldetag:
Date of filing:
Date de dépôt:
14/02/03

Anmelder:
Applicant(s):
Demandeur(s):
AGFA-GEVAERT
2640 Mortsel
BELGIUM

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Method for outputting a digital product definition of a printed product to a layout application

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	EP	Tag: Date: Date:	21/11/02	Aktenzeichen: File no. Numéro de dépôt:	EPA 21026133

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing:
Etats contractants désignés lors du depôt:
AT/BG/BE/CH/CY/CZ/DE/DK/EE/ES/FI/FR/GB/GR/HU/IE/IT/LI/LU/MC/

Bemerkungen:
Remarks:
Remarques:

[ABSTRACT]

METHOD FOR OUTPUTTING A DIGITAL PRODUCT DEFINITION OF A PRINTED PRODUCT TO A LAYOUT APPLICATION

A method for making a digital representation of a printed product, the method including (a) creating by a planning application a product definition of the printed product and (b) outputting the product definition by the planning application to a layout application for using the product definition by the layout application for creating an artwork for making the digital representation of the printed product.

[DESCRIPTION]

TERMS

Digital Representation of a Printed Product: the product that is to be printed, represented in digital form.

Artwork: the electronic document that is generated by the Designer, created while taking into account the Product Definition. Also called design, layout or publication.

Designer: person in the creative department who will create an artwork in electronic format based on specific instructions and content.

Product Definition: definition for the final printed product, holding manufacturing specifications. Some specifications that are important for a designer are: page size, bleed size, number of pages, colors used, etc. A product definition often starts from a product intent.

Scripting: a way to automate operations in an application. Could be used to preset and/or lock some attributes in a layout application.

Layout application: software application that is used by the Designer to generate the artwork. Common layout applications are InDesign from Adobe, Xpress from Quark, Illustrator from Adobe, FreeHand from Macromedia.

Template:

A good example is a Word template. It is a special type of document that is set up with the proper information. When the designer starts to create the artwork, he asks the layout application to create a new document based on this template.

For some applications, the template is limited to one (or two facing) page(s). In this case, the exact number of pages cannot be controlled.

Do not confuse with Imposition templates.

Starting document: similar to the template but it is now already a 'real' document. The application can 'Open' it directly and start working into it.

Template Generator: a software application that creates a template to be used by a layout application, based on certain parameters (of the product definition).

Locked (template or artwork): crucial attributes of a template or document may be locked so that the Designer cannot modify them. E.g. page and bleed size, number of pages, color model, spot colors. Remark: the terms "attributes" and "parameters", e.g. of the product definition, are used as synonyms.

Planning system, also called Planning Application, Project Management System or MIS: a system that creates the product definition. This product definition is to be used both by the creative department, that creates the artwork, and by the pre-press department, that sets up a pre-press workflow for the printed product. Generally, the product definition is also used by the finishing and the printing departments.

Pre-press workflow application: the software application that is used to generate the pre-press product.

PROBLEM

The designer needs to create the artwork according to the product definition. Later, the pre-press department will set up a workflow system also according to the product definition.

Errors can occur:

- when transferring the instructions from a planning system to the designer;
- when the designer applies the instructions to the artwork;
- when the pre-press department gets instructions in a different way than the designer.

This is especially a problem because the creative department and the pre-press department are really "two different worlds" ("creative" people versus "production" people). Often, the creative and the pre-press departments belong to different companies.

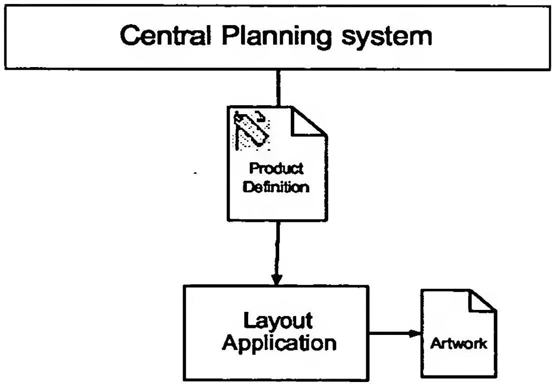
POSSIBLE SOLUTION

- Make sure that the same product definition is used both by the designer, in the creative department, and by the pre-press department. It is preferred that a central planning system (covering both creative department and pre-press department) is used to define the product BEFORE the creative work is started. In this way, no errors are introduced when transferring the product definition to another department.
- Create the basic document automatically, based on this product definition.
- Preferably lock at least one, more preferably lock all crucial attributes.
- Practical Examples:
- Example 1: Agfa Delano (as the planning application)
  - o creates the product definition in JDF format;
  - o sends the product definition to Adobe InDesign (as the layout application);
  - o sends the product definition to Agfa Apogee (as the pre-press workflow application).
- Example 2: Adobe InDesign
  - o creates a new document according to the product definition;
  - o locks the attributes to avoid errors.
- Remark: in example 1, "sending" the product definition from Agfa Delano to Adobe InDesign means that Agfa Delano, the planning application, outputs the product definition and that Adobe InDesign, the layout application, inputs it. In this document, "inputting" and "outputting" may be either directly (from the first application to the second one) or indirectly (from the first to the second application via an intermediate application - this is illustrated e.g. by the third example under the heading "Implementation" below).

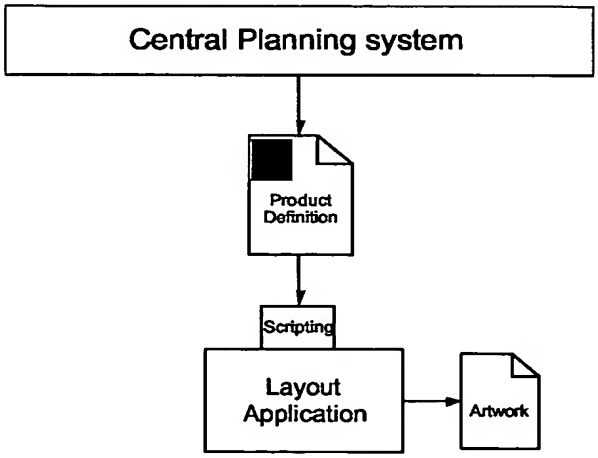
IMPLEMENTATION

There are several possible solutions depending on what the layout application supports, ordered here with the most integrated solution first. As mentioned above, preferably a central planning system (covering both creative department and pre-press department) is used; such a planning system is indicated in the drawings below. However, a planning system that creates the product definition and that only sends it to the layout application, not to the pre-press workflow application, may also be used in the embodiments illustrated by the drawings below.

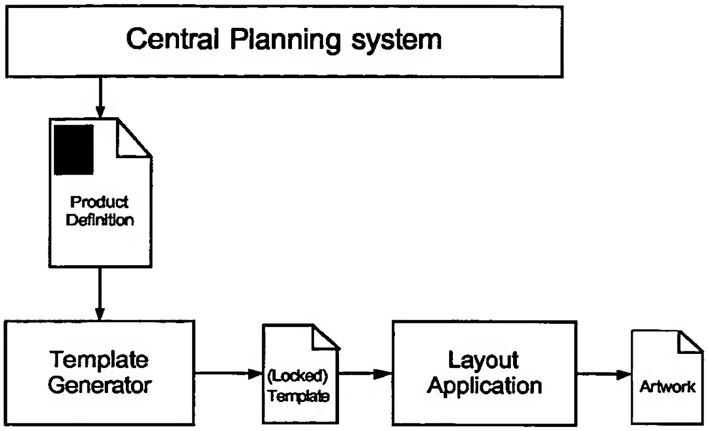
1. Send the product definition (parameters) in a format that the layout application understands.



2. Send the product definition (parameters) to a script ("an intermediate script") that drives the layout application to produce the starting document (or template).



3. Send the product definition (parameters) to an intermediate application that generates the document template or starting document.



The following example is a script, written in Visual Basic, for InDesign; it illustrates the embodiment of the second drawing above.

EXAMPLE

```
' version 0.99
' by Koen Van de Poel
' Copyright Agfa-Gevaert N.V.
' 2002-12-7
'
' Sample script for Indesign that allows to Import a JDF with
Product Intents
' set-up a UI with it and then create an Indesign or Illustrator
Document with the required settings
'
'Option Explicit
Dim xmldoc As DOMDocument


' Creates the Indesign document
Private Sub Indesign_Click()
    Dim myInDesign As Indesign.Application
    Dim myDocument As Indesign.Document
    Dim myTextFrame As Indesign.TextFrame

    Set myInDesign = CreateObject("Indesign.Application.2.0")

    Rem maak nieuwe pub
    Set myDocument = myInDesign.Documents.Add
    Rem copy from interface
    myDocument.DocumentPreferences.NumberOfPages =
CInt(NrPages.Text)
    myDocument.DocumentPreferences.PageWidth = PageWidth.Text & "mm"
    myDocument.DocumentPreferences.PageHeight = PageHeight.Text &
"mm"
    ' myDocument.Name = Title.Text
    Set myMetaDataPreferences = myDocument.MetaDataPreferences
    myMetaDataPreferences.Author = "KVDP Indesign JDF Script"
    myMetaDataPreferences.JobName = JobName.Text
    myMetaDataPreferences.Description = Description.Text
    myMetaDataPreferences.Title = Title.Text

Rem just to show something
    Set myTextFrame = myDocument.Spreads.Item(1).TextFrames.Add
    myTextFrame.GeometricBounds = Array("10mm", "10mm", "30mm",
"30mm")
    myTextFrame.TextContents = "Hello World"
End Sub
' Similar stuff for Illustrator automation
Private Sub IllustratorButton_Click()
    Dim appRef As New Illustrator.Application
    Dim newDoc As Illustrator.Document
    Dim newCMYKColor As New Illustrator.CMYKColor
    Dim newColor As New Illustrator.Color
    Dim newSpot As Illustrator.Spot
    Dim frontPath As Illustrator.PathItem

     Rem copy from interface
    'myDocument.DocumentPreferences.NumberOfPages =
CInt(NrPages.Text)
    'myDocument.DocumentPreferences.PageWidth = PageWidth.Text &
"mm"
    'myDocument.DocumentPreferences.PageHeight = PageHeight.Text &
"mm"
    ' myDocument.Name = Title.Text
    'Set myMetaDataPreferences = myDocument.MetaDataPreferences
    'myMetaDataPreferences.Author = "KVDP Indesign JDF Script"
    'myMetaDataPreferences.JobName = JobName.Text
    'myMetaDataPreferences.Description = Description.Text
    'myMetaDataPreferences.Title = Title.Text
    'AppActivate "Adobe Illustrator"
    'Set appRef = CreateObject("Illustrator.Application.10.0")

    ' Create the document
    Set newDoc = appRef.Documents.Add(aiDocumentCMYKColor,
(CInt(PageWidth.Text) * 72) / 25.4, (CInt(PageHeight.Text) * 72) /
25.4)

    ' Create a spot color (as test)
    newCMYKColor.cyan = 22
    newCMYKColor.yellow = 100
    newCMYKColor.black = 33
    newCMYKColor.magenta = 44

    newColor.CMYK = newCMYKColor

    Set frontDocument = appRef.ActiveDocument
    Set newSpot = frontDocument.Spots.Add
    newSpot.Color = newColor
    newSpot.ColorType = aiSpot
    newSpot.name = "Koens Spot"

End Sub
Rem Import JDF
Private Sub initxmldoc()
    Set xmldoc = Nothing
    Set xmldoc = CreateObject("Msxml.DomDocument")
    xmldoc.async = False
End Sub

' Load the JDF document into internal structure
Private Function LoadJDF(xmldoc As DOMDocument, filepath As String)
As Boolean

    Dim runlistnode As IXMLDOMNode
    Dim bLoadOK As Boolean
    Dim Title As String
    bLoadOK = False

    If Not (IsNull(xmldoc)) Then
        xmldoc.Load filepath ' load XML file as document
        ' quick validity check by looking for Runlist

        If IsValidNode(xmldoc.documentElement) Then
            Set runlistnode =
xmldoc.documentElement.selectSingleNode(kSizeIntentDimensions)
            If Not IsValidNode(runlistnode) Then
                Dim mResult As VbMsgBoxResult
                ' allow to proceed, should be better with Proceed
stop buttons !!
                mResult = MsgBox("This JDF file is not supported (no
SizeIntent found)." & Chr(10) & "Use supplied templates." & Chr(10)
& filepath & Chr(10) & _
                "Click 'Yes' to Proceed anyway and 'No' to stop.", _
                vbYesNo + vbDefaultButton2 + vbCritical)
                If (mResult = vbNo) Then
                    bLoadDefault = False
                End If
            Else: bLoadOK = True
            End If
            If bLoadOK Then
                ' check if file has proper version
                Dim vers As String
                'vers = GetJDFVersion(xmldoc)
                'If Not ((vers = JDFVersion10) Or (vers =
JDFVersion11)) Then
                '    MsgBox "The version of this JDF file (version:"
& vers & ") is not supported." & Chr(10) & "Use supplied templates."
& filepath, vbOKOnly + vbCritical
                '    bLoadDefault = True
               ' End If
            End If
        Else ' invalid document element
            MsgBox "Could not open JDF " & filepath, vbOKOnly +
vbCritical
            bLoadOK = False
        End If
    End If
    LoadJDF = bLoadOK
End Function

Private Sub Command1_Click()

End Sub

Rem Import JDF
' This function allows the user to select a JDF file (triggered by
clicking the ImportJDF button in the UI)
Private Sub ImportJDF_Click()
    Dim FileName As String

    ' InitXML
    Set xmldoc = Nothing
    Set xmldoc = CreateObject("Msxml.DomDocument")
    xmldoc.async = False

    Dim filepath As Variant
    Dim strFilter As String
    Dim lngFlags As Long

    ' Prepare Select File dialog box
    strFilter = ahtAddFilterItem(strFilter, "JDF Files (*.jdf)",
"*.jdf")
    strFilter = ahtAddFilterItem(strFilter, "All Files (*.*)",
"*.*")
    filepath = ahtCommonFileOpenSave(InitialDir:="", _
        Filter:=strFilter, FilterIndex:=1, flags:=lngFlags, _
        DialogTitle:="Open", hwnd:=Me.hwnd)

    ' Extract the data needed from the JDF file
    If (filepath <> vbNullString) Then
        FileName = filepath
        ' Load the file
        If LoadJDF(xmldoc, FileName) Then
          ' Extract the data and copy to locals
          Description.Text = GetJDFDescriptiveName(xmldoc)
          JobName.Text = GetJDFJobID(xmldoc)
          NrPages.Text = GetJDFIntentPages(xmldoc)
          JDFDimensions = GetJDFIntentDimensions(xmldoc)
          ' Convert to mms
          sPos = InStr(JDFDimensions, " ")
          Wstr = Left(JDFDimensions, sPos - 1)
          Hstr = Mid(JDFDimensions, sPos + 1)
          W = Round((CDbl(Wstr) / 72) * 25.4)
          PageWidth.Text = W
          H = Round((CDbl(Hstr) / 72) * 25.4)
          PageHeight.Text = H
        End If

    End If
End Sub
```

Those skilled in the art will appreciate that numerous modifications and variations may be made to the embodiments disclosed above without departing from the scope of the present invention.

✍

[CLAIMS]

1. A method for making a digital representation of a printed product comprising the steps of:
   - creating by a planning application a product definition of said printed product;
   - outputting said product definition by said planning application to a layout application for using said product definition by said layout application for creating an artwork for making said digital representation of said printed product.

2. The method according to claim 1 further comprising the steps of:
   - outputting said product definition by said planning application to a pre-press workflow application for using said product definition by said pre-press workflow application for creating a pre-press product.

3. A method for creating an artwork for making a digital representation of a printed product, the method comprising the steps of:
   - inputting a product definition of said printed product by a layout application from a planning application;
   - using said product definition for creating said artwork by said layout application.

4. The method according to any one of the preceding claims wherein said product definition comprises a first parameter selected from the group of a page size, a number of pages, a bleed size and a set of colors used.

5. The method according to any one of the preceding claims further comprising the step of:
   - locking a second parameter of said product definition for protecting said second parameter from being modified by said layout application.

6. The method according to any one of claims 1 to 5 further comprising the step of:
   - creating said product definition in Job Definition Format, i.e. JDF.

7. The method according to any one of claims 1 to 5 further comprising the step of:
   - outputting said product definition by said planning application to an intermediate script for driving said layout application.

8. The method according to any one of claims 1 to 5 further comprising the step of:
   - outputting said product definition by said planning application to an intermediate application for sending said product definition in a particular format to said layout application.

9. A data processing system comprising means for carrying out the steps of the method according to any one of claims 1 to 8.

10. A computer program comprising computer program code means adapted to perform the method according to any one of claims 1 to 8 when said program is run on a computer.

11. A computer readable medium comprising program code adapted to carry out the method according to any one of claims 1 to 8 when run on a computer.

ø